ELSEVIER

Theoretical Computer Science 285 (2002) 89–99

Theoretical Computer Science

www.elsevier.com/locate/tcs

The hardness of placing street names in a Manhattan type map ☆

Sebastian Seibert *, Walter Unger

Lehrstuhl für Informatik I, RWTH Aachen, D-52056 Aachen, Germany

Accepted May 2001

Abstract

Map labeling is a classical key problem. The interest in this problem has grown over the last few years, because of the need to churn out different types of maps from a growing and altering set of data. We will show that the problem of placing street names without conflicts in a rectangular grid of streets is NP-complete and APX-hard. More precisely, we show that there is no polynomial time $(\frac{176}{175}-\varepsilon)$ approximation algorithm unless P = NP. This is the first result of this type in this area. Further importance of this result arises from the fact that the considered problem is a simple one. Each row and column of the rectangular grid of streets contains just one street and the corresponding name may be placed anywhere in that line. © 2002 Published by Elsevier Science B.V.

1. Introduction and definitions

The street layout of some modern cities planned on a drawing table are often right angled. In a drawing of such a city's map, the street names should be placed without conflict. Thus, each name should be drawn within the rectangular area without splitting or conflicting with other names. See Fig. 1 for an example. The names are indicated by simple lines.

A *map* consists of an $N_h \times N_v$ grid with N_h columns and N_v rows. A *horizontal line*, i.e. a horizontal street name, is given by the pair (i,l), where i $(1 \leqslant i \leqslant N_v)$ indicates the row of the street and l $(1 \leqslant l \leqslant N_h)$ the length of that name. The *vertical lines* are given in a similar way, by indicating their column and height.

A *placement* in a map assigns to every line (street name) a position to place the first character in. If the first character of a horizontal line (i,l) is placed in column

☆ Supported by DFG Project HR 14/5-1 "Zur Klassifizierung der Klasse praktisch lösbarer algorithmischer Aufgaben".

* Corresponding author.
E-mail address: seibert@i1.informatik.rwth-aachen.de (S. Seibert).

0304-3975/02/$ - see front matter
PII: S0304-3975(01)00292-4

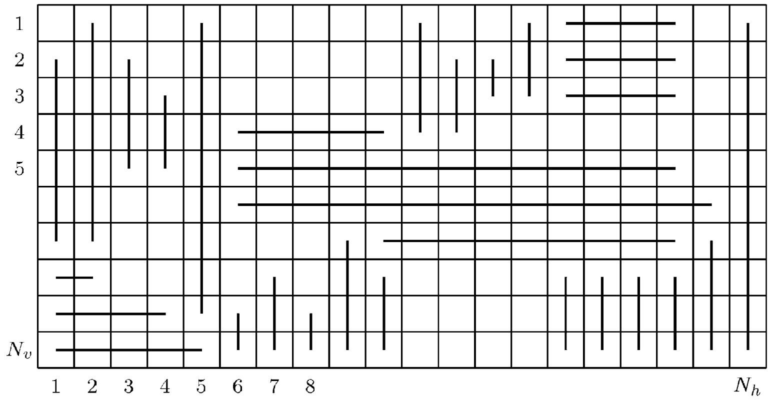

Fig. 1. A map with street names.

s $(1 \leqslant s \leqslant N_{\mathrm{h}} - l + 1)$, then the name will occupy the following space in the grid: $\{(j, i) \mid s \leqslant j \leqslant s + l - 1\}$. Vertical lines are placed analogously. A *conflict* in a placement is a position occupied by both, a vertical and a horizontal line.

Given a map and a set of lines to be placed in the map, StrP denotes the problem to decide whether the given lines may be placed conflict-free. We will show that this problem is NP-complete. Max-StrP is the problem to find a placement that maximizes the number of lines placed without conflict, which will be shown to be APX-hard below. This, we will do by showing the NP-hardness of $(1, \frac{175}{176} + \varepsilon)$-StrP, for $\varepsilon > 0$, the restriction of StrP to inputs where either all or at most a fraction of $\frac{175}{176} + \varepsilon$ of the street names can be placed conflict free at a time. [1] An overview on approximation problems and hardness of approximation proofs can be found in [5, 9, 1].

Map labeling problems have been investigated intensively in the last 20 years. Most results are in the field of heuristic and practical implementation [14]. The known complexity results are about restricted versions of the map labeling problem. In [7, 3, 10, 8, 6], NP-completeness results are presented for the case that each label has to be attached to a fixed point. The type of label and alignment varies in these papers. For other models of the map labeling problem see [14].

Note that the type of map labeling studied here was motivated by practical applications and introduced in [11]. In that paper, there are algorithms given for solving StrP which run for some special cases in a polynomial time and perform reasonably in practical applications. With this respect, the APX-hardness is even more surprising. For the harder problem to place lines on a cylindrical map, there is also a proof of NP-completeness in [11] which relies essentially on the cylindrical structure.

[1] Note that in the preliminary version of this paper [12], contained in the proceedings of CIAC 2000, the lower bound on the approximability of Max-StrP was given as $\frac{224}{223} - \varepsilon$. We have been able to improve the bound since, by a slight modification of the construction.

Note that our notation differs from [11] where for instance a vector of street-lengths for all rows and columns is given as input. However, for showing APX-hardness by a reduction from 3SAT, we need to describe efficiently a map of the order $n^2 \times n$ having only $\mathcal{O}(n)$ lines to be placed, if n is the size of the formula.

Our reduction is based on the following result. Note that in the formulae constructed in the proof of that theorem, every variable is used at least twice. [2]

Theorem 1 (Håstad [4]). *For every small $\varepsilon > 0$, the problem to distinguish* 3SAT *instances that can be satisfied from those where atmost a fraction of $\frac{7}{8} + \varepsilon$ of the clauses can be satisfied at the same time, $(1, \frac{7}{8} + \varepsilon) - 3SAT$ for short, is* NP-*hard.*

The construction and proof of NP-completeness of StrP will be presented in Section 2. Section 3 contains the proof for the APX-completeness of Max-StrP and Section 4 gives the conclusions.

2. Map construction and NP-hardness

In this section, we give a reduction from 3SAT to StrP which we use for showing NP-hardness as well as APX-hardness. The proof of NP-hardness comes immediately with the construction. For the proof of APX-hardness of StrP only a small construction detail had to be added. Thus, we give in this section the full construction, and we prove the APX-hardness of StrP in the next section.

Assume that we have a 3SAT formula ϕ consisting of clauses $c_1, \ldots, c_l$ over variables $x_1, \ldots, x_m$. Each clause c_i is of the form $z_{i,1} \vee z_{i,2} \vee z_{i,3}$, the literals $z_{i,j}$ being from $\{x_1, \ldots, x_m\} \cup \{\overline{x_1}, \ldots, \overline{x_m}\}$. We assume w.l.o.g. that each variable occurs at least twice. Moreover, we assume that there are at least as many positive occurrences of a variable as there are negative. This can be assured by simply renaming variables (exchanging x with $\bar{x}$).

First we give an informal description of the proof idea. We need to construct a map M_ϕ out of ϕ. Let $n = 3l$. M_ϕ will have height $N_{\mathrm{v}} = 14n$ and width $N_{\mathrm{h}} \leqslant 36n^2$. It will be split into several vertical stripes, which means that there are several vertical lines of full height N_{v}. For clear reference, we use names for the stripes. The general picture of M_ϕ is shown in Fig. 2.

It consists of three groups of vertical stripes. To the left, we have one vertical stripe for each variable. Here, the placement of lines corresponds to assigning a certain value to the respective variable. The rightmost part contains for each clause a triple of vertical stripes such that a non-conflicting placement of all lines in these stripes can be made only if there is a clause satisfying the setting of at least one of its variables represented in the leftmost part. The middle part, called "mirror negative occurrences" is necessary to connect the other two properly. It will be explained below.

[2] This is not surprising at all: variables used at most once in a formula can be assigned a truth value trivially, hence their inclusion would rather make the problem more tractable, not harder.

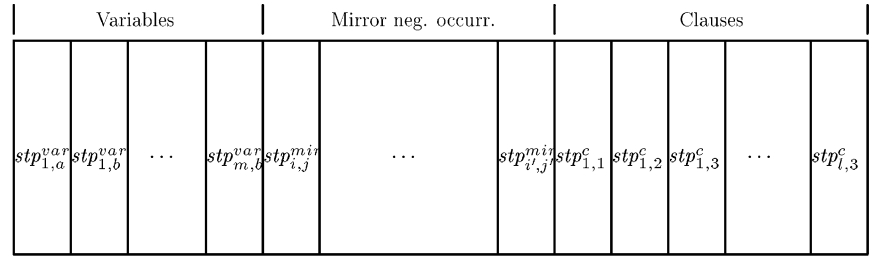

Fig. 2. The outline of the map.

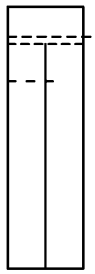

Fig. 3. A vertical stripe.

To ease the description we will just define the width of these stripes and further lines added within the stripes. Thus, the position of the separating vertical lines will be implicitly defined. The width of these stripes varies between $6n$ and $4n - l > 3n$. The position of vertical lines put within such a stripe will be given relative to the separating vertical lines. Let *stp* be a stripe of width w surrounded by vertical lines at position c and $c + w + 1$. If we say a vertical line *is put at position* i $(1 \leqslant i \leqslant w)$ *in stripe stp*, this line will be put in column $c + i$ of the whole map.

To move the information around in M_ϕ, our main tools are the horizontal lines which fit into the above mentioned stripes. These horizontal lines will be ordered (from top and bottom rows towards the middle) according to their length. A line of width $6n - j$ will be in row $n + j$ or $N_v - n - j$, where $0 \leqslant j < 3n$. The reason for leaving the top and bottom n rows unused will become clear below. We denote by $H(j)$ a horizontal line of width $6n - j$ put in row $n + j$ and by $\bar{H}(j)$ a horizontal line of width $6n - j$ put in row $N_v - n - j$.

In the middle of a typical vertical stripe of width $6n - j$, we will put a vertical line of height $N_v - 1 - (n + j)$. Consequently, either $H(j)$ or $\bar{H}(j)$ can be placed in this stripe, by placing the middle vertical line either at the lowest position, opening row $n + j$ for $H(j)$, or at the highest position, opening row $N_v - n - j + 1$ for $\bar{H}(j)$. Furthermore, no other horizontal line $H(j')$ or $\bar{H}(j')$ $(0 \leqslant j' < 3n,\ j' \neq j)$ can be placed in that stripe. Either it is wider than the whole stripe, or if it is smaller, it will be in a row which is always blocked by the middle vertical line, see Fig. 3.

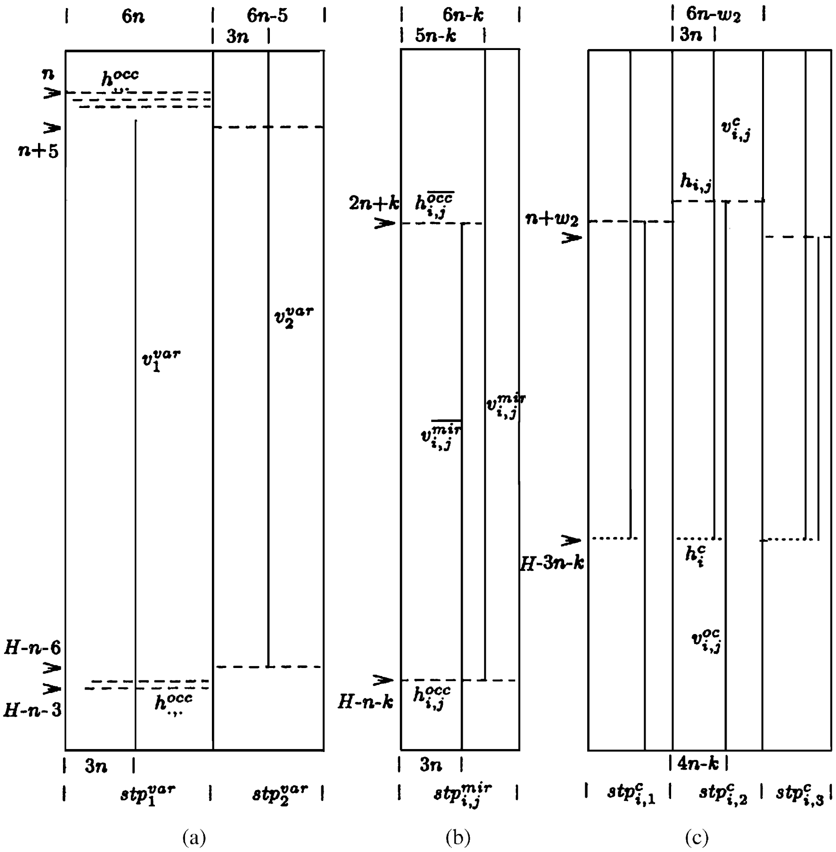

Fig. 4. Components of the map.

There will be some exception where we may place several horizontal lines in the same stripe. But as we will see, the overall principle will remain unaffected.

The details of the construction are shown in Fig. 4. Here, we have solid lines depicting vertical lines drawn in their definite position or in one of their only two possibly non-conflicting positions. These positions are always limited to the respective vertical stripe. Dashed lines are used for the horizontal lines which are the essential tool to connect the different parts of the map. They always have two possibly non-conflicting positions in two different parts of the map. For some of them, you can see both positions in Fig. 4. Finally, there is the dotted line shown in all of its three possibly non-conflicting positions in the rightmost part.

2.1. The variable part

More precisely, we start constructing M_ϕ by building a vertical stripe stp_i^{var} for each variable x_i ($1 \leqslant i \leqslant m$), as shown in Fig. 4(a) for the first two variables. Assume that x_i

occurs s_i times in ϕ. Furthermore, let $t_i = \sum_{j=1}^{i-1} s_j$, and $t_0 = 0$. Stripe stp_i^{var} is set to the width $w_i = 6n - t_{i-1}$. You will see that $s_1 = 5$ and $s_2 = 2$ gives a picture as in Fig. 4(a). Note that in Fig. 4, numbers above and below the map give the amount of free space between vertical lines, whereas numbers to the left mark the row a horizontal line is put in.

Now we put a vertical line v_i^{var} in the middle of the stripe stp_i^{var} to prevent unwanted placement of horizontal lines. Since all we need is that at most $3n$ columns are left free to either side (all horizontal lines will be wider), it only has to be roughly the middle. Thus, we put vertical lines of height $N_{\text{v}} - n - t_i$ in stripe stp_i^{var} at the respective position $3n + 1$.

Next, we add a horizontal line for each occurrence of the variable x_i. If $z_{j,j'}$ contains the kth occurrence of x_i in ϕ, then $h_{j,j'}^{\text{occ}}$ represents that occurrence. If $z_{j,j'} = x_i$, then $h_{j,j'}^{\text{occ}} = H(t_{i-1} + k)$. On the other hand, if $z_{j,j'} = \overline{x_i}$, then $h_{j,j'}^{\text{occ}} = \bar{H}(t_{i-1} + k)$. These are the dashed lines of Fig. 4(a).

Now, the fact that the first and last n rows of M_ϕ that are not used for horizontal lines takes effect. The height of v_i^{var} is always at least $N_{\text{v}} - 2n$. Thus, it is impossible to have horizontal lines placed both below *and* above v_i^{var} at the same time. Consequently, from any placement of v_i^{var}, changing it into at least one of its extremal positions is safe.

Thus, we may always assume that v_i^{var} is either placed topmost or bottommost, i.e. it builds a switch effectively. This is depicted in Fig. 4(a) for v_1^{var}, and v_2^{var}, respectively. Here, it is unimportant as to which placement exactly a horizontal line $h_{j,j'}^{\text{occ}}$ gets in stripe stp_i^{var}.

If the switch is placed bottommost (as shown for v_1^{var}), then all lines corresponding to positive occurrences of x_i can be placed without conflict in stripe stp_i^{var}. The lines corresponding to negative occurrences cannot be placed without conflict in stripe stp_i^{var}. This placement will correspond to setting $x_i = 1$. Similarly, the other extremal placement of the switch (as shown for v_2^{var}) corresponds to setting $x_i = 0$.

We will end up using just $n = 3l$ different widths of horizontal lines, from $6n$ to $5n + 1$. Thus, the next step can be started by using width $5n$.

2.2. *The middle part*

The aim of the middle part is to mirror the horizontal lines representing the negative occurrences of variables into the upper half of the map. This is done by using a stripe $stp_{i,j}^{\text{mir}}$ as depicted in Fig. 4(b). Assume that the occurrence under consideration is $z_{i,j} = \overline{x_u}$.

The new stripe $stp_{i,j}^{\text{mir}}$ has the width of the horizontal line to be mirrored. Let that line be $h_{i,j}^{\text{occ}} = \bar{H}(k)$, $0 \leqslant k < n$. The width of stripe $stp_{i,j}^{\text{mir}}$ is $6n - k$. Now we add inside the stripe a vertical line $v_{i,j}^{\text{mir}}$ at position $(5n - k) + 1$ of height $N_{\text{v}} - 1 - (n + j)$. This reduces the width of the stripe for all inner rows from $6n - k$ to $5n - k$. Inside this reduced stripe, we put new lines $h_{i,j}^{\overline{\text{occ}}} = H(n + k)$ and a vertical line $v_{i,j}^{\overline{\text{mir}}}$ at position $3n + 1$ of height $N_{\text{v}} - 1 - (2n + j)$.

Let us look at the idea of this construction. If x_u is set to 0 (i.e. the line v_u^{var} is placed topmost), then $h_{i,j}^{\mathrm{occ}}$ can be placed in stp_u^{var}. Then $v_{i,j}^{\overline{\mathrm{mir}}}$ can be placed in the lowest position. This again allows to place $h_{i,j}^{\overline{\mathrm{occ}}}$ above it.

If x_u is set to 1, then $h_{i,j}^{\mathrm{occ}}$ must be placed in stripe $stp_{i,j}^{\mathrm{mir}}$. Then both $v_{i,j}^{\mathrm{mir}}$ and $v_{i,j}^{\overline{\mathrm{mir}}}$ are "pushed upwards" which makes it impossible to place $h_{i,j}^{\overline{\mathrm{occ}}}$ in $stp_{i,j}^{\mathrm{mir}}$ without conflict.

The entire middle part of the map is made up of stripes like that. Note that the new horizontal lines are all of different widths as were the original ones. We just reduce the width by n.

2.3. Stripes for clauses

Finally, we go into constructing a triple of stripes $stp_{i,1}^{\mathrm{c}}$, $stp_{i,2}^{\mathrm{c}}$ and $stp_{i,3}^{\mathrm{c}}$ for each clause c_i. Let $c_i = z_{i,1} \vee z_{i,2} \vee z_{i,3}$, and let $4n-k > 3n$ be a new width, not used before. Consider the horizontal lines $h_{i,1}, h_{i,2}, h_{i,3}$ representing those occurrences in the upper half of rows. That means $h_{i,j} = h_{i,j}^{\mathrm{occ}}$ if $z_{i,j}$ is a positive occurrence of a variable, and $h_{i,j} = h_{i,j}^{\overline{\mathrm{occ}}}$ if it is a negative occurrence (for $j \in \{1,2,3\}$). For speaking about the width of the new stripes, let $h_{i,j} = H(w_j)$ for $j \in \{1,2,3\}$.

For each of the three occurrences, we construct a stripe $stp_{i,1}^{\mathrm{c}}$, $i \in \{1,2,3\}$ analogously to those described for the middle part, except that we use as new line width for all three stripes the same value $4n-k$. Consequently, only one new horizontal line $h_i^{\mathrm{c}} = \bar{H}(2n+k)$ is created. The width of the stripe $stp_{i,j}^{\mathrm{c}}$ is w_j for $j \in \{1,2,3\}$.

Each of the three stripes will have new vertical lines that are $v_{i,j}^{\mathrm{c}}$ of height $N_{\mathrm{v}} - 1 - n - w_j$ at position $4n-k+1$ (reducing the width to $4n-k$) and $v_{i,j}^{\bar{\mathrm{c}}}$ of height $N_{\mathrm{v}} - 1 - 3n - k$ at position $3n$ (the "middle" line), see Fig. 4(c).

Now the overall effect of this is as follows. Assume one of the literals $z_{i,j}$ in a clause c_i is $z_{i,j} = x_u$, and x_u is set to 1, represented by placing the lines of the variable switch for x_u as described above. Then the corresponding horizontal line $h_{i,j} = h_{i,j}^{\mathrm{occ}}$ can be placed in stp_u^{var}. This leaves free the place of that line in the corresponding stripe $stp_{i,j}^{\mathrm{c}}$ of the clause. The vertical lines $v_{i,j}^{\mathrm{c}}, v_{i,j}^{\bar{\mathrm{c}}}$ of that stripe can be placed in the highest position, and finally, the unique horizontal line h_i^{c} can be placed in that stripe. Similarly, assume that $z_{i,j} = \overline{x_u}$, and x_u is set to 0, represented by placing the lines of the variable switch for x_u as described above. Then $h_{i,j}^{\mathrm{occ}}$ can be placed in stp_u^{var}, and $h_{i,j} = h_{i,j}^{\overline{\mathrm{occ}}}$ can be placed in $stp_{i,j}^{\mathrm{mir}}$ after placing $v_{i,j}^{\mathrm{mir}}$ and $v_{i,j}^{\overline{\mathrm{mir}}}$ in the lowest position. Again, this leaves free the place of line $h_{i,j}$ in the corresponding stripe $stp_{i,j}^{\mathrm{c}}$ of the clause, and finally, h_i^{c} can be placed in that stripe.

If on the other hand none of the three variables is set to fulfill the clause, then that causes all three horizontal lines $h_{i,1}, h_{i,2}, h_{i,3}$ to be placed in their respective stripes $stp_{i,1}^{\mathrm{c}}, stp_{i,2}^{\mathrm{c}}, stp_{i,3}^{\mathrm{c}}$. However, this "pushes down" the vertical lines $v_{i,1}^{\mathrm{c}}$, $v_{i,1}^{\bar{\mathrm{c}}}$, $v_{i,2}^{\mathrm{c}}$, $v_{i,2}^{\bar{\mathrm{c}}}$, $v_{i,3}^{\mathrm{c}}$, $v_{i,3}^{\bar{\mathrm{c}}}$ which results in the impossibility to place the clause representing line h_i^{c} conflict free.

Overall, each non-satisfied clause corresponds to an unavoidable conflict. So far, we have shown our first result.

Theorem 2. StrP *is* NP-*hard.*

3. A bound of approximability

In this section, we will show that the previous construction can be used to obtain thresholds on approximating StrP.

Theorem 3. *For every small* $\varepsilon > 0$, *the problem to distinguish* StrP *instances that can be placed conflict-free from those where atmost a fraction of* $\frac{175}{176} + \varepsilon$ *of the lines can be placed without conflict*, $(1, \frac{175}{176} + \varepsilon)-$StrP *for short*, *is* NP-*hard.*

Corollary 1. *For every small* $\varepsilon > 0$, *there is no polynomial time* $(\frac{176}{175} - \varepsilon)$*-approximation algorithm for* Max-StrP *unless* $P = \mathrm{NP}$, *i.e.* Max-StrP *is* APX-*hard.*

Proof of Theorem 3. In view of Theorem 1, it is sufficient to show that the above construction satisfies the following claims.

1. From a formula ϕ containing l clauses, the above construction yields, in polynomial time, a map M_ϕ containing atmost $22l$ lines (if every variable is used at least twice).
2. There exists a polynomial time procedure P_a that works on a map M_ϕ as follows. Given a placement p where m *horizontal* lines have conflicts, P_a outputs a placement p', where at most m horizontal lines have conflicts, such that each has only one conflict. Moreover, all horizontal lines with conflicts are of the type h_i^c constructed in the last part of the above construction.
3. There exists a polynomial time procedure P_b that works on a map M_ϕ as follows. Given a placement p' as generated by P_a with m conflicts, P_b generates an assignment to the variables of ϕ such that at most m clauses of ϕ are not satisfied.

Then, assuming we would have an algorithm A deciding $(1, \frac{175}{176} + \varepsilon)-$StrP in polynomial time, we could get one deciding $(1, \frac{7}{8} + 22\varepsilon)-$3SAT in polynomial time as follows. (Note that we do not need the fact that P_a and P_b are efficient.)

Given $(1, \frac{7}{8} + 22\varepsilon)-$3SAT instance ϕ with l clauses, construct M_ϕ and apply the assumed decision algorithm.

If ϕ is satisfiable, then there is a conflict-free placement in M_ϕ as we have already seen in Section 2.

If on the other hand at most a fraction $\frac{7}{8}+22\varepsilon$ of the clauses of ϕ wherever satisfiable, then we look at M_ϕ. Assume p to be a placement where a maximal number of lines in M_ϕ are placed without conflict. Let m be the number of horizontal lines having a conflict under placement p. That is, the fraction of lines which could be placed without conflict in M_ϕ is at most $(22l - m)/22l$.

Now we apply procedures P_a and P_b, getting an assignment satisfying at least $l - m$ out of l clauses in ϕ. By assumption about ϕ, we have

$$\frac{l-m}{l} \leqslant \frac{7}{8} + 22\varepsilon \quad \text{that is} \quad m \geqslant \left(\frac{1}{8} - 22\varepsilon\right) l$$

and hence

$$\frac{22l-m}{22l} \leqslant \frac{22l-(\frac{1}{8}-22\varepsilon)l}{22l} = \frac{175}{176}+\varepsilon.$$

Thus, the result of algorithm A would decide $(1, \frac{7}{8}+22\varepsilon)$−3SAT, in contradiction to Theorem 1.

It remains to prove the above claims.

1. As mentioned in Section 2, there are at most $\frac{3}{2}l$ variables if we assume every variable to be used at least twice. Furthermore, we have exactly $3l$ occurrences of variables and l clauses.
   Constructing the leftmost part of M_ϕ, we have invented 2 lines per variable and 1 per occurrence which gives at most $6l$ lines. In the middle part, we need 4 new lines per negative occurrence, that is at most $6l$ lines (remember that at most half of the occurrences are negative). Finally, in the rightmost part of M_ϕ, we use 9 new vertical and one new horizontal line per clause, being $10l$ new lines.
   Overall, there are at most $22l$ lines to be placed in M_ϕ.
2. We use the names of the lines and stripes as given in Section 2.
   The basic idea of P_a is that each horizontal line is best placed in one of the vertical stripes "made for it". More precisely, P_a works by modifying p as follows.
   (a) For each variable x_u, place the line v_u^{var} either topmost or bottommost. Which one of the two possibilities is taken, is decided in such a way that the minimal number of conflicts between v_u^{var} and those lines $h_{i,j}^{\mathrm{occ}}$ (placed as in p) occurs where $z_{i,j}$ is an occurrence of x_u, and $h_{i,j}^{\mathrm{occ}}$ is placed completely within stripe stp_u^{var}.
   (b) Place every line $h_{i,j}^{\mathrm{occ}}$ which still has a conflict in stripe $stp_{i,j}^{\mathrm{c}}$ if $z_{i,j}$ is a positive occurrence, and in $stp_{i,j}^{\mathrm{mir}}$ if $z_{i,j}$ is a negative occurrence.
   (c) For each negative occurrence $z_{i,j}$, place $v_{i,j}^{\mathrm{mir}}$ and $v_{i,j}^{\overline{\mathrm{mir}}}$ in their highest position if $h_{i,j}^{\mathrm{occ}}$ is placed in $stp_{i,j}^{\mathrm{mir}}$, and in lowest position otherwise.
   (d) If $v_{i,j}^{\mathrm{mir}}, v_{i,j}^{\overline{\mathrm{mir}}}$ are placed in the highest position, place $h_{i,j}^{\overline{\mathrm{occ}}}$ in $stp_{i,j}^{\mathrm{c}}$. Otherwise, place $h_{i,j}^{\overline{\mathrm{occ}}}$ in $stp_{i,j}^{\mathrm{mir}}$.
   (e) If $h_{i,j}$ is placed in stripe $stp_{i,j}^{\mathrm{c}}$, place $v_{i,j}^{\mathrm{c}}$ and $v_{i,j}^{\bar{\mathrm{c}}}$ in their lowest position, otherwise in their highest position.
   (f) It h_i^{c} still has a conflict, place it in any of the stripes $stp_{i,1}^{\mathrm{mir}}$, $stp_{i,2}^{\mathrm{mir}}$, or $stp_{i,3}^{\mathrm{mir}}$.

   The crucial point is that none of these steps increase the number of horizontal lines having conflicts. We check this step by step.
   (a) Due to the general principle of the construction, all horizontal lines other than those $h_{i,j}^{\mathrm{occ}}$ the representing occurrences of x_u cannot be placed without conflict within the stripe for x_u. As we have seen in Section 2, the minimal number of conflicts between v_u^{var} and the lines representing occurrences of x_u is assumed in one of the extremal positions of v_u^{var}. Consequently, the change of placing v_u^{var} can only decrease the number of conflicts.
   (b) Only horizontal lines already having conflicts are affected.

(c) Only $h_{i,j}^{\mathrm{occ}}$ and $h_{i,j}^{\overline{\mathrm{occ}}}$ may be placed in $stp_{i,j}^{\mathrm{mir}}$ without conflict. If $h_{i,j}^{\mathrm{occ}}$ is present, this step may move a conflict from $h_{i,j}^{\mathrm{occ}}$ to $h_{i,j}^{\overline{\mathrm{occ}}}$. Otherwise, only a possible conflict of $h_{i,j}^{\overline{\mathrm{occ}}}$ is resolved.
(d) Again, only horizontal lines having conflicts are affected.
(e) Analogously to step (c), a conflict may only be moved from $h_{i,j}$ to h_i^{c}.
(f) Again, only horizontal lines already having conflicts are affected.

This guarantees that the number of horizontal lines already having conflicts is not increased.

Moreover, steps (b) and (c) for negative occurrences, resp. (b) and (e) for positive occurrences, assure that lines $h_{i,j}^{\mathrm{occ}}$ are placed without conflict. Remember that $h_{i,j} = h_{i,j}^{\mathrm{occ}}$ for positive occurrences, and $h_{i,j} = h_{i,j}^{\overline{\mathrm{occ}}}$ for the negative. Steps (d) and (e) guarantee the same for $h_{i,j}^{\overline{\mathrm{occ}}}$.

Finally, the lines h_i^{c} are placed in a stripe where they may have a conflict with at most some $v_{i,j}^{\bar{\mathrm{c}}}$, i.e. have atmost one conflict.

3. In Section 2, we have convinced ourselves that the “variable switches” in the leftmost part of M_ϕ can be placed without conflict only in two ways, interpretable as setting the corresponding variable to 0 or 1. Since all conflicts are on the rightmost part of M_ϕ, we can take that interpretation as an assignment to the variables. Also in Section 2, we have seen that this assignment satisfies a clause c_i iff the line h_i^{c} can be placed without conflict. Thus, if there were initially m conflicts present in the placement, at most m clauses will be non-satisfied. □

4. Conclusion

We have presented the first proofs of NP-completeness and APX-hardness for a simple map labeling problem. An algorithm with an approximation factor of two is easy. Just label either all horizontal or all vertical streets. It remains open to close the gap between both factors. Our results extend easily to the case where some regions of the map may not be used for labels. It is also interesting to look for more extensions.

References

[1] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, M. Protasi, Complexity and Approximation, Springer, Berlin, 1999.
[2] J. Christensen, J. Marks, S. Shieber, Labeling point features on maps and diagrams, Harvard CS, 1992, TR-25-92.
[3] M. Formann, F. Wagner, A packing problem with applications to lettering of maps, Proc. 7th Annu. ACM Symp. Comput. Geom., 1991, pp. 281–288.
[4] J. Håstad, Some optimal inapproximability results, Proc. 29th Ann. ACM Symp. Theory Comput. 1997, pp. 1–10.
[5] D.S. Hochbaum (Ed.), Approximation Algorithms for NP-hard Problems, PWS Publishing Company, Boston, 1996.
[6] Claudia Iturriaga, Anna Lubiw, NP-Hardness of Some Map Labeling Problems, University of Waterloo, Canada, 1997, CS-97-18.

[7] T. Kato, H. Imai, The NP-completeness of the character placement problem of 2 or 3 degrees of freedom, Record of Joint Conf. of Electrical and Electronic Engineers in Kyushu, 1988, pp. 1138.
[8] D.E. Knuth, A. Raghunathan, The problem of compatible representatives, SIAM J. Discrete Math. 5 (3) (1992) 422–427.
[9] E.W. Mayr, H.J. Prömel, A. Steger (Eds.), in: Lectures on Proof Verification and Approximation Algorithms, Lecture Notes in Computer Science, 1967, Springer, Berlin, 1998.
[10] J. Marks, S. Shieber, The computational complexity of cartographic label placement, TR-05-91, Harvard CS, 1991.
[11] G. Neyer, F. Wagner, Labeling downtown, in: G.C. Bongiovanni, G. Gambosi, R. Petreschi (Eds.), Algorithms and Complexity, Proc. 4th Italian Conf. CIAC 2000, Lecture Notes in Computer Science, 1767, Springer, Berlin, 2000, pp. 113–124.
[12] S. Seibert, W. Unger, The hardness of placing street names in a manhattan type map, in: G.C. Bongiovanni, G. Gambosi, R. Petreschi (Eds.), Algorithms and Complexity, Proc. 4th Italian Conf. CIAC 2000, Lecture Notes in Computer Science, 1767, Springer, Berlin, 2000, pp. 102–112.
[13] T. Strijk, A. Wolff, Labeling Points with Circles, Institut für Informatik, Freie Universität Berlin, 1999, B 99-08.
[14] T. Strijk, A. Wolff, The map labeling bibliography, http://www.inf.fu-berlin.de/map-labeling/bibliography.